Class Project

MET CS 665 - Software Design and Patterns
(20 points)
GitHub Invitation Link
`https://classroom.github.com/a/k6UaVFa9`

1 Project Guidelines

One of the requirements for CS665 course is the final project (20% of the grade). The project must be done individually. This is an opportunity for you to be creative in solving a problem that is of interest to you and demonstrate your software design proficiency. The project should be challenging enough so that you could discuss with professional software developers.

We have covered in our class a list of design patterns. We have learned about *Strategy, Factory, Abstract Factory, Singleton, Prototype, Command, Observer, State, Template, Facade, Decorator, Composite, Adapter, Proxy, Iterator and Mediator patterns*.

However, the list of existing design patterns are longer than what we covered in our class. You can find a list of design patterns that can be used in Java programming language on the following website:
`http://java-design-patterns.com/patterns/`

Your main task for the class project is to learn about further design patterns. You should pick up one of the design patterns that we have not covered in our class, learn it on your own, create and describe a use case scenario for it (it should be a new scenario which is not described in any references like books or Websites), implement it in your favorite programing language and create a presentation about the design pattern and your scenario.

It is recommended to can use in your project a combination of design patterns because in real software projects mostly a combination of patterns are used.

Your project should include one design pattern that we have not covered in our class. You read the above list and find select one of the design patterns that you like. Some examples of important patterns that we have not covered deeply in our class are: *Visitor, Bridge, Mediator, Builder, Callback, Delegation, Thread Pool*.

Note: Your project idea should not be copied from existing books or Websites. Your project should be a unique project based on your own ideas.

2 Project Tasks

2.1 Task 1 : Design pattern and its Use Case Scenario Description. (4 points)

Provide a description of the application use case scenario that you have selected for your final project.

Describe what are your main software design concepts regarding this application.
For example describe:

- What are the design goals in your project?
- How is the flexibility, of your implementation, e.g., how you add or remove in future new types?
- How is the simplicity and understandability of your implementation?
- How you avoided duplicated code?

You should use design patterns in your project. You can use a combinations of
you have used any design patterns, describe which design pattern you have applied and why.
Write your description in a **README.md** file, use MarkDown format `https://spec.commonmark.org/current/` and add the README.md file to the root folder of your project.

2.2 Task 2 - UML Class Diagram. (5 points)

Create a class model for your application, containing 5-8 of the most important classes. It should encompass the functionality of the use case application described above. Show only non-obvious and key methods.

2.3 Task 4 - Implement your solution in Java (5 points)

- You should use the provided project templates to implement your project.
- Provide a zip file that includes your implementation package.

  The zip file should include a **README.md** file that describes how to compile and run your implementation. Create a Zip file from the root of your working directory so that it includes all subdirectories of your project. Remove the binary files before creating the zip files.
- Your zip file should **not be larger that 10MB**, include only source files of your project but not the binaries that you generated.
- Document your code very well. The best way to write documentation inside your code is to write the code while you are implementing the project and not postpone it to later time.
- You should use the Google Java Style Guide
  (`https://google.github.io/styleguide/javaguide.html`)

  For C++ Implementation you use Google C++ Style Guide
  `https://google.github.io/styleguide/cppguide.html`
- We should be compile your solution on a unix-based operating system like Linux or MacOS. Your project should include a **run.sh** that can be used to compile and run your project.

2.4 Task 4 - Create a presentation about your project (5 points)

Create a presentation for your final project and upload a ppt, pdf or odp file.

2.5 Task 5 - Record your presentation and upload a MP4 file. (5 points)

Present your talk on your desktop and record it using a desktop recording software. You can also use your webcam and add it to your presentation.

You can use Kaltura Capture Recorder. `https://onlinecampus.bu.edu/bbcswebdav/courses/00cwr_odeelements/metcs/cs_Kaltura.htm`

Note: We recommend to use Kaltura and BU my-media system. However, if you have any issues with that and can not use it, you can record your presentation using ZOOM client application, generate a MP4 file, upload your MP4 file to some clouds (like google Drive, DropBox) and share with us the Link of your presentation.

2.6 Using GitHub Classroom

For all of your assignments you should use GitHub Classroom.

You should use the provided project templates to implement your project. Login to your github account and click on the Github Invitation Link, a fresh private project with propagated template project will be created for you.

Follow the following steps:

1. You need to have your own GitHub account.
2. Login to your own GitHub Account
3. Click on the Invitation Link for this Assignment and Click on Accept to accept the assignment. You can find the GitHub Link for this assignment on the first page of the assignment.
4. You need to provide access to your GitHub Classroom in our account (This is a one time setup for the GitHub Classroom App).
5. Then a private repository will be created for you that you, your instructor and facilitator have access to it.
6. You should use ”Star” the assignment repository to be able to find it easier and use it when you develop your assignment.
7. A good rule of thumb is commit your code to the repository as soon as you have implemented some small piece of code that does something and can compile.

2.7 Turnin

1. When you are finished with your assignment. You need to download all of your code from GitHub or have the latest version of your code on your computer.
2. Create a single document that has results for all three tasks. For example a PDF document for UML diagrams.
3. Please zip up all of your code and your document (use .zip only, please!). Remember to remove the binary files, these are normally in bin/ or target/ folders. The binary files can increase the size of your zip file.
4. Double check if you have uploaded correctly your zip file. You can download back your file, unzip it and check if it is the correct file and it is correctly zipped. We will evaluate the zip file that you have uploaded to the blackboard and cannot evaluate wrong or damaged files.

You can use the green download button of GitHub, download a zip file of your repository and upload it to blackboard.

Please note that we will grade your final zip file uploaded to the Blackboard, but we will also check the history of your GitHub repository. Both versions should be the same. The main reason why we want to have zip file on blackboard is to archieve a zip file of your assignment on blackboard.

2.8 Grading

You solution should be a self-contained solution that can be compiled executed based on the instruction given in your README.md file. We recommend to use our project templates, and add your implementation to our template and use build tools like maven or sbt. If your solution program is complete based on the requested functionalities, can compile and run then you would get the full points.

We will grade your solution, and reduce the points for each task based on the following grading policy.

- Your UML diagram does not include important components like Interfaces/Classes - 5% reduction for each component.
- We will compile all solution using *"mvn compile"* command in your project (we will download your zip file, unzip it and run the "mvn compile" - your project should compile using Java JDK 1.8). If your code does not compile for any reason it will cause 10% grade deduction for the implementation task.
- Your code includes functionality bugs - 10% deduction for each bug
- Your code should include a README.md that describes your conceptual solution, how to compile and execute. If your program does not include such README.md file, or your README file does not include all requested infos then we will reduce 10% of points.
- Your program does not implement the requested functionalities - 10% deduction for each functionality.
- We will programmatically check all solutions for plagiarism using jplag (https://github.com/jplag/jplag) (We will run some scripts on your codes). If your code is a exact duplicate of someone else's solution, then we can not accept your solution. We will contact you regarding the issue.

2.9 Assignment Completion and Late Work

All assignments should be submitted on time. If there is a delay, the student must be in touch with the instructor and his/her facilitator.

Late submissions without reasons will result in grade deduction. You can turn in an assignment up to **24 hours late, in which case you receive a 10% penalty** (that is, 10 points are subtracted from an assignment that is worth 100 points), or **up to 48 hours late, in which case you receive a 20% penalty**. Assignments turned in after (more than 48 hours late) that are **not accepted**.

We kept on saying **no exceptions, but there are exceptions** in very extreme circumstances, with proper documentation. For example, if you obtain a doctor/dentist note stating that you were

so ill at the due date/time that you could not reasonably be expected to meet the deadline, it is possible to get an extension.

2.10 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between ”cheating” and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way — visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**.

As far as going to the web and using Google, we will apply the **”two line rule”**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the ”two line rule” inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you’re looking at something you wrote a while ago and you need to remind yourself what you were thinking.